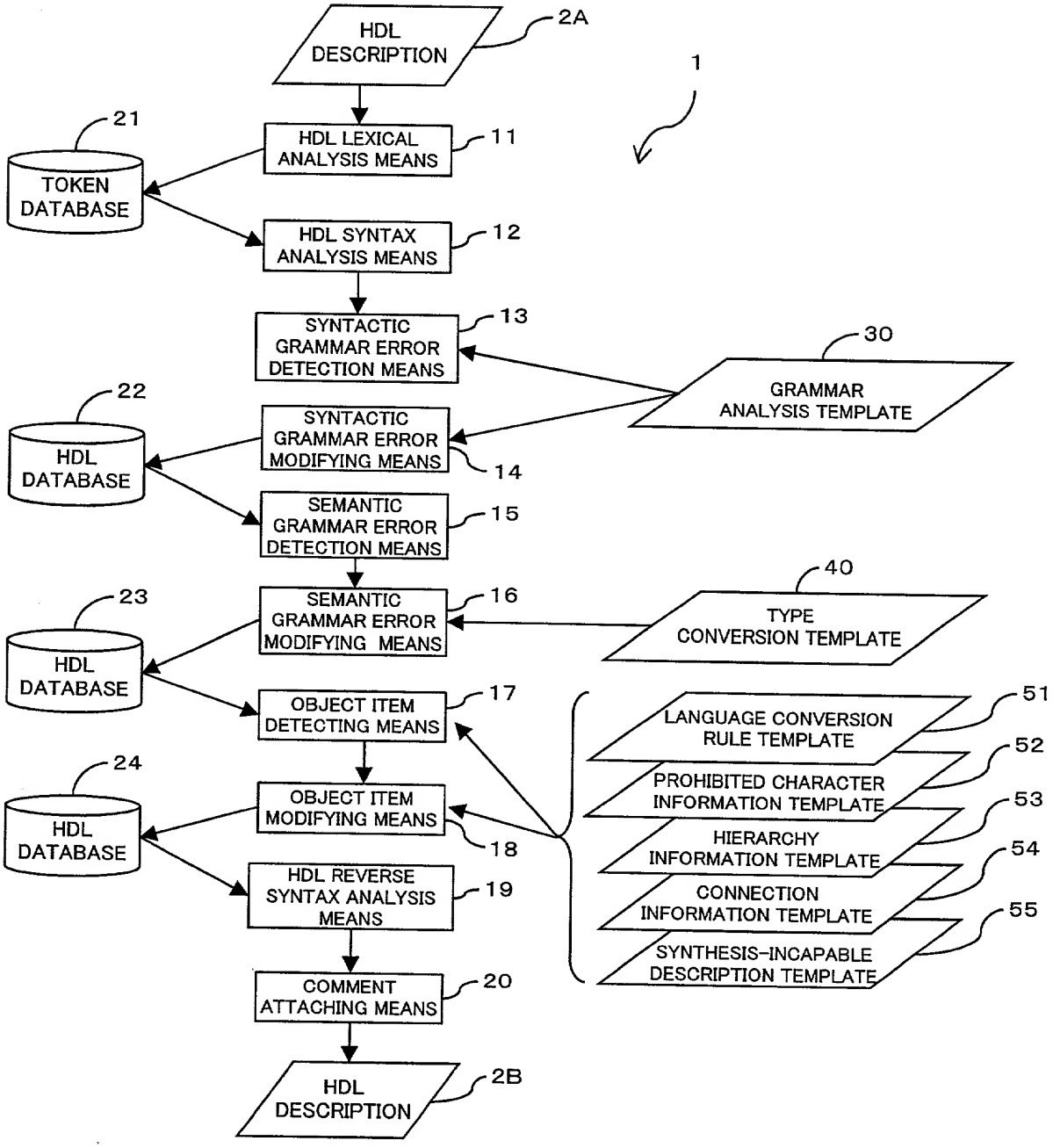
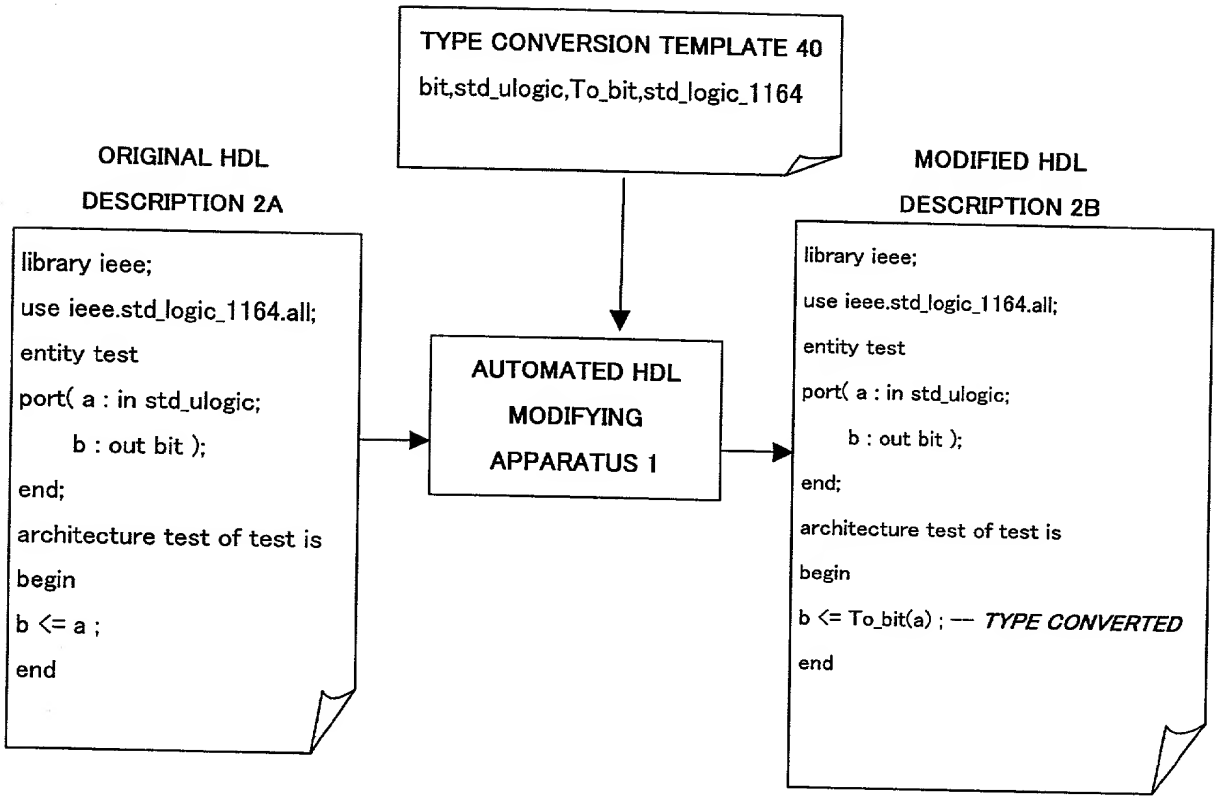
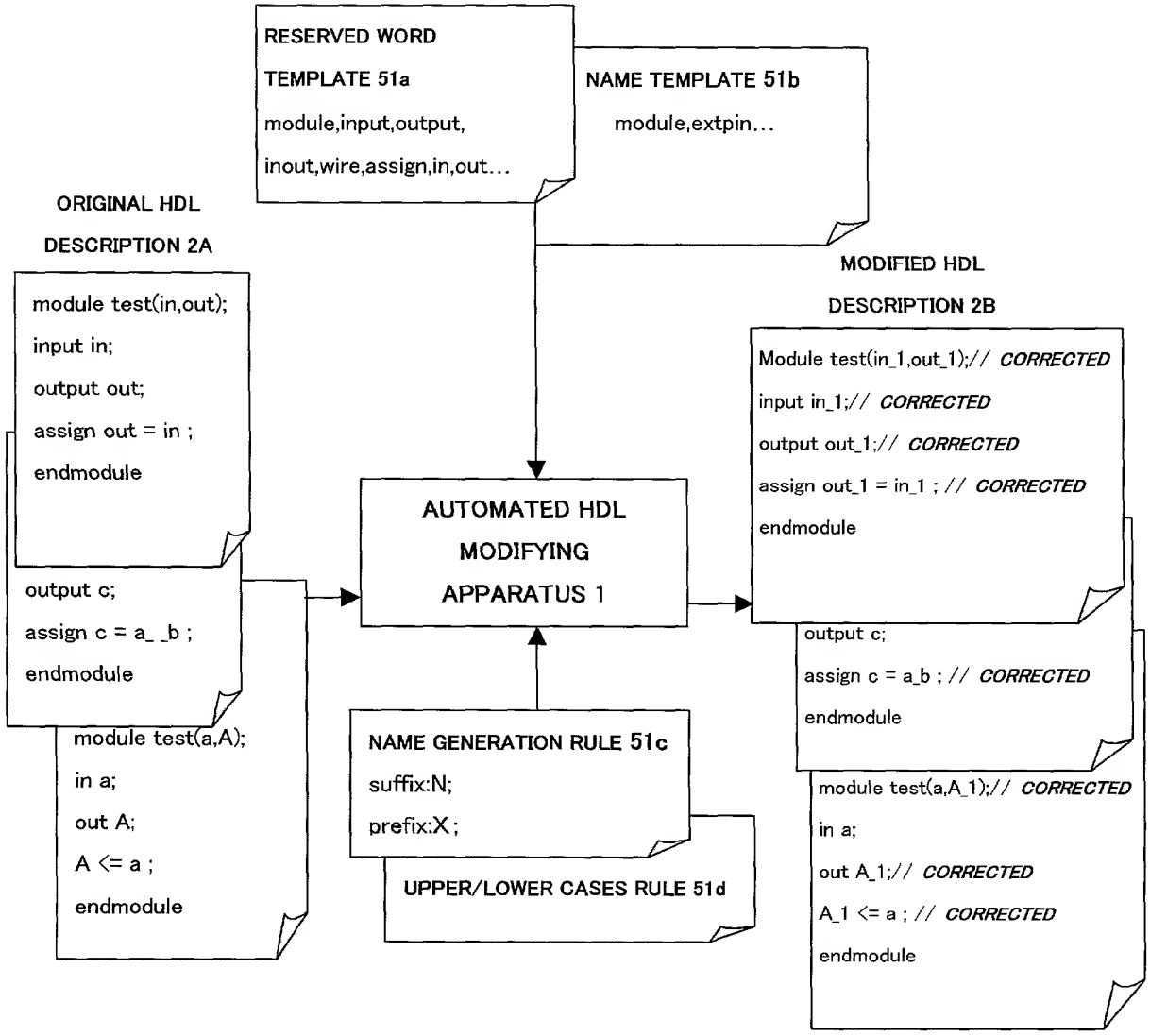
# FIG. 1


HDL DESCRIPTION
2A
1
21
HDL LEXICAL ANALYSIS MEANS
11
TOKEN DATABASE
HDL SYNTAX ANALYSIS MEANS
12
SYNTACTIC GRAMMAR ERROR DETECTION MEANS
13
30
GRAMMAR ANALYSIS TEMPLATE
22
HDL DATABASE
SYNTACTIC GRAMMAR ERROR MODIFYING MEANS
14
SEMANTIC GRAMMAR ERROR DETECTION MEANS
15
40
23
SEMANTIC GRAMMAR ERROR MODIFYING MEANS
16
TYPE CONVERSION TEMPLATE
HDL DATABASE
OBJECT ITEM DETECTING MEANS
17
51
LANGUAGE CONVERSION RULE TEMPLATE
52
PROHIBITED CHARACTER INFORMATION TEMPLATE
53
HIERARCHY INFORMATION TEMPLATE
54
CONNECTION INFORMATION TEMPLATE
55
SYNTHESIS-INCAPABLE DESCRIPTION TEMPLATE
24
HDL DATABASE
OBJECT ITEM MODIFYING MEANS
18
HDL REVERSE SYNTAX ANALYSIS MEANS
19
COMMENT ATTACHING MEANS
20
HDL DESCRIPTION
2B

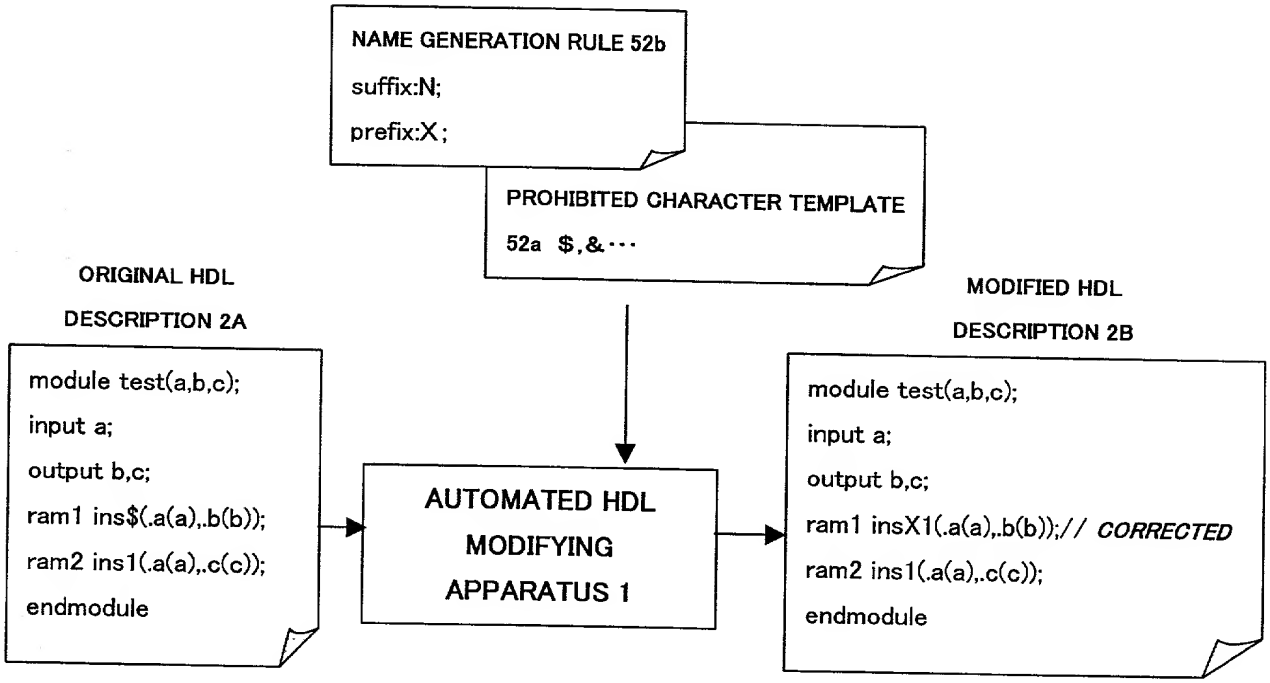
# FIG. 2


TYPE CONVERSION TEMPLATE 40
bit,std_ulogic,To_bit,std_logic_1164
ORIGINAL HDL DESCRIPTION 2A
library ieee;
use ieee.std_logic_1164.all;
entity test
port( a : in std_ulogic;
b : out bit );
end;
architecture test of test is
begin
b <= a ;
end
AUTOMATED HDL MODIFYING APPARATUS 1
MODIFIED HDL DESCRIPTION 2B
library ieee;
use ieee.std_logic_1164.all;
entity test
port( a : in std_ulogic;
b : out bit );
end;
architecture test of test is
begin
b <= To_bit(a) ; -- TYPE CONVERTED
end

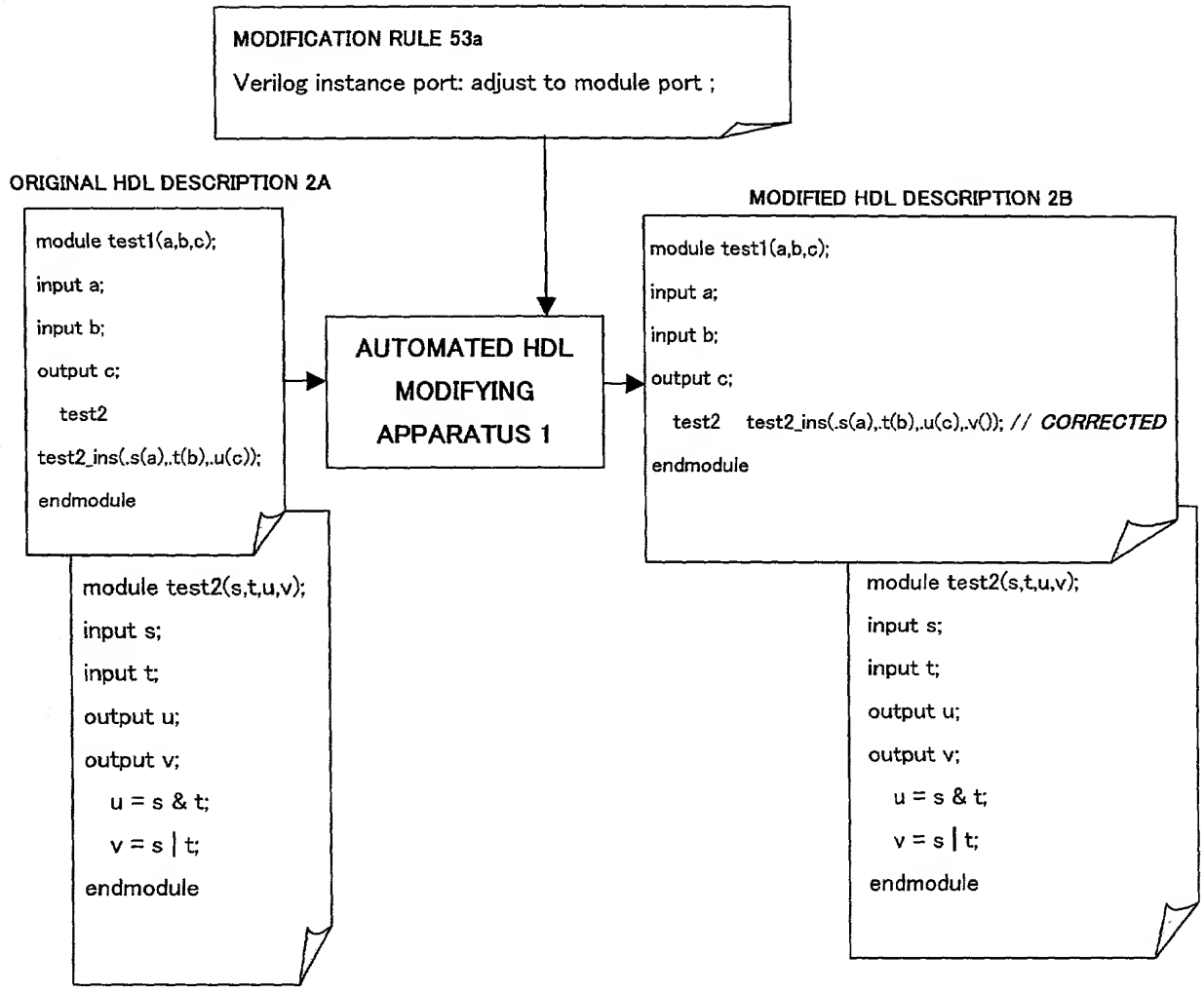
# FIG. 3


RESERVED WORD
TEMPLATE 51a
module,input,output,
inout,wire,assign,in,out...
NAME TEMPLATE 51b
module,extpin...
ORIGINAL HDL
DESCRIPTION 2A
module test(in,out);
input in;
output out;
assign out = in ;
endmodule
output c;
assign c = a_ _b ;
endmodule
module test(a,A);
in a;
out A;
A <= a ;
endmodule
AUTOMATED HDL
MODIFYING
APPARATUS 1
NAME GENERATION RULE 51c
suffix:N;
prefix:X;
UPPER/LOWER CASES RULE 51d
MODIFIED HDL
DESCRIPTION 2B
Module test(in_1,out_1);// CORRECTED
input in_1;// CORRECTED
output out_1;// CORRECTED
assign out_1 = in_1 ; // CORRECTED
endmodule
output c;
assign c = a_b ; // CORRECTED
endmodule
module test(a,A_1);// CORRECTED
in a;
out A_1;// CORRECTED
A_1 <= a ; // CORRECTED
endmodule

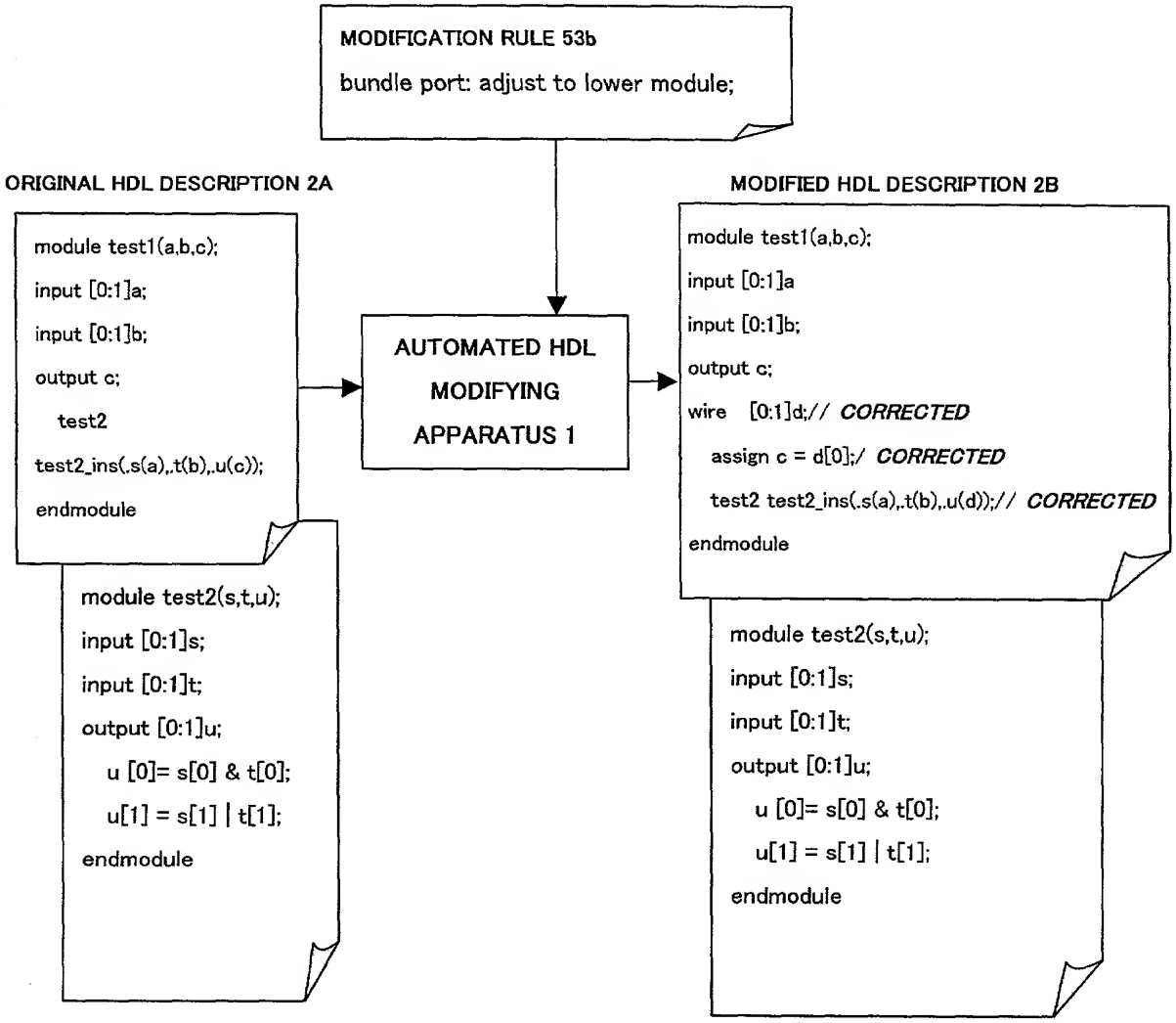
# FIG. 4


NAME GENERATION RULE 52b
suffix:N;
prefix:X;
PROHIBITED CHARACTER TEMPLATE
52a $,&・・・
ORIGINAL HDL
DESCRIPTION 2A
module test(a,b,c);
input a;
output b,c;
ram1 ins$(.a(a),.b(b));
ram2 ins1(.a(a),.c(c));
endmodule
AUTOMATED HDL
MODIFYING
APPARATUS 1
MODIFIED HDL
DESCRIPTION 2B
module test(a,b,c);
input a;
output b,c;
ram1 insX1(.a(a),.b(b));// CORRECTED
ram2 ins1(.a(a),.c(c));
endmodule

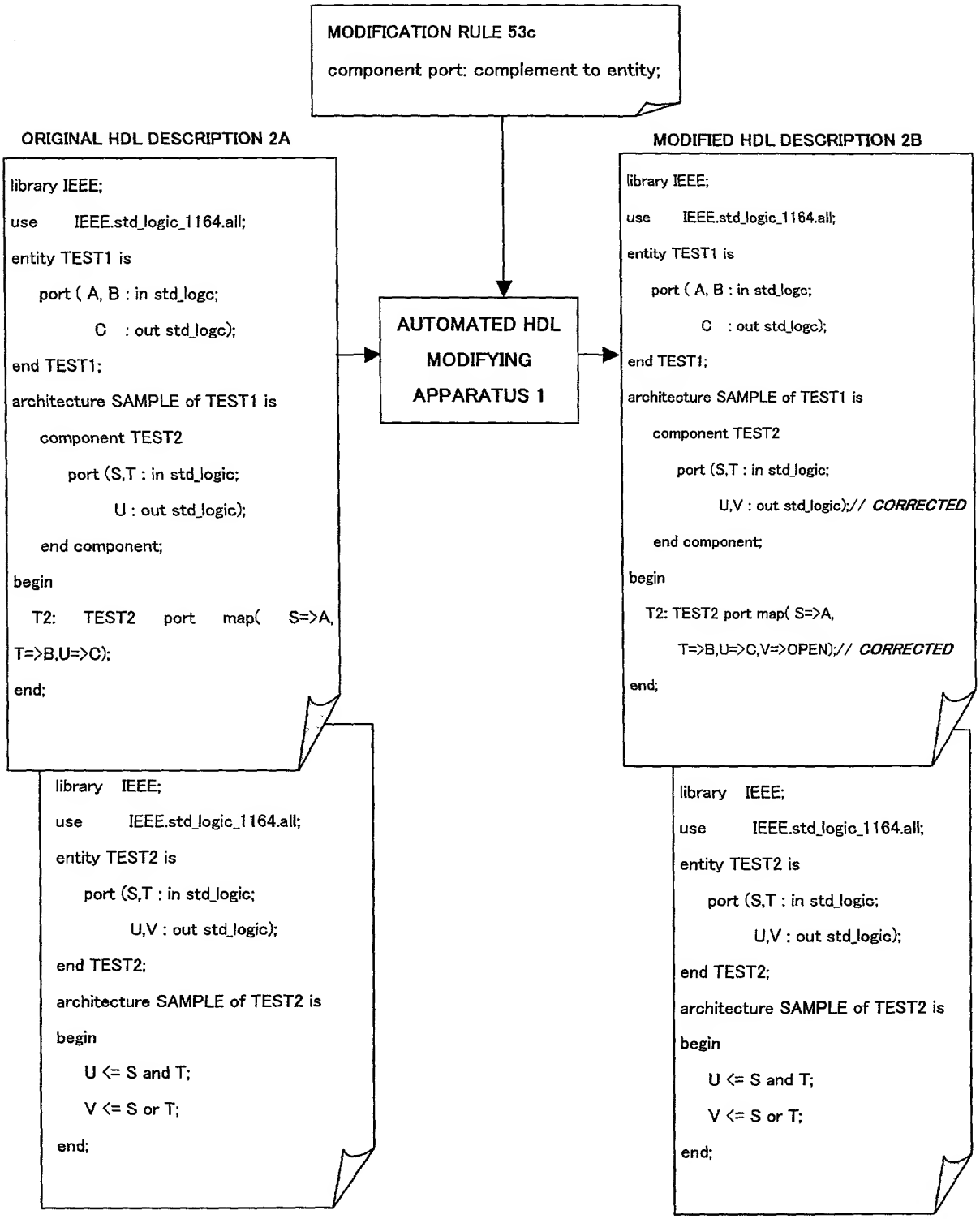
# FIG. 5


MODIFICATION RULE 53a
Verilog instance port: adjust to module port ;
ORIGINAL HDL DESCRIPTION 2A
module test1(a,b,c);
input a;
input b;
output c;
test2
test2_ins(.s(a),.t(b),.u(c));
endmodule
module test2(s,t,u,v);
input s;
input t;
output u;
output v;
u = s & t;
v = s | t;
endmodule
AUTOMATED HDL MODIFYING APPARATUS 1
MODIFIED HDL DESCRIPTION 2B
module test1(a,b,c);
input a;
input b;
output c;
test2 test2_ins(.s(a),.t(b),.u(c),.v()); // CORRECTED
endmodule
module test2(s,t,u,v);
input s;
input t;
output u;
output v;
u = s & t;
v = s | t;
endmodule

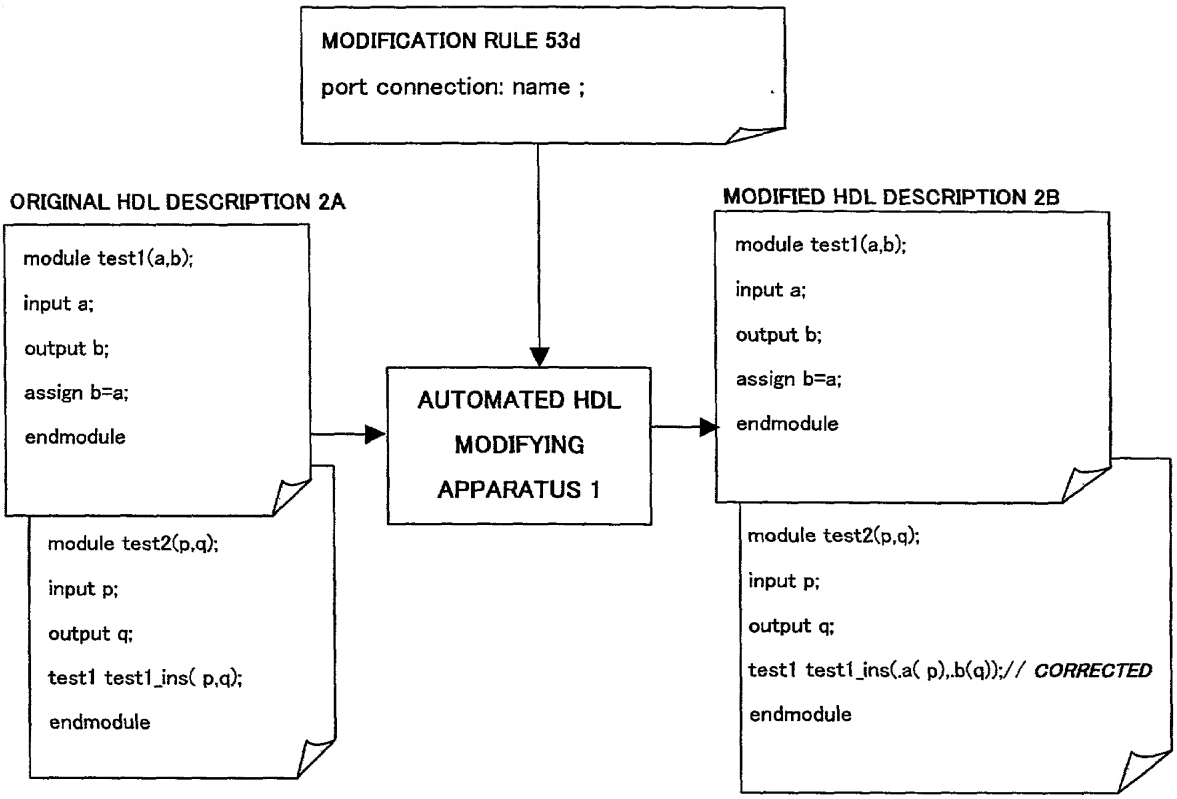
# FIG. 6


MODIFICATION RULE 53b
bundle port: adjust to lower module;
ORIGINAL HDL DESCRIPTION 2A
module test1(a,b,c);
input [0:1]a;
input [0:1]b;
output c;
test2
test2_ins(.s(a),.t(b),.u(c));
endmodule
module test2(s,t,u);
input [0:1]s;
input [0:1]t;
output [0:1]u;
u [0]= s[0] & t[0];
u[1] = s[1] | t[1];
endmodule
AUTOMATED HDL MODIFYING APPARATUS 1
MODIFIED HDL DESCRIPTION 2B
module test1(a,b,c);
input [0:1]a
input [0:1]b;
output c;
wire [0:1]d;// CORRECTED
assign c = d[0];/ CORRECTED
test2 test2_ins(.s(a),.t(b),.u(d));// CORRECTED
endmodule
module test2(s,t,u);
input [0:1]s;
input [0:1]t;
output [0:1]u;
u [0]= s[0] & t[0];
u[1] = s[1] | t[1];
endmodule

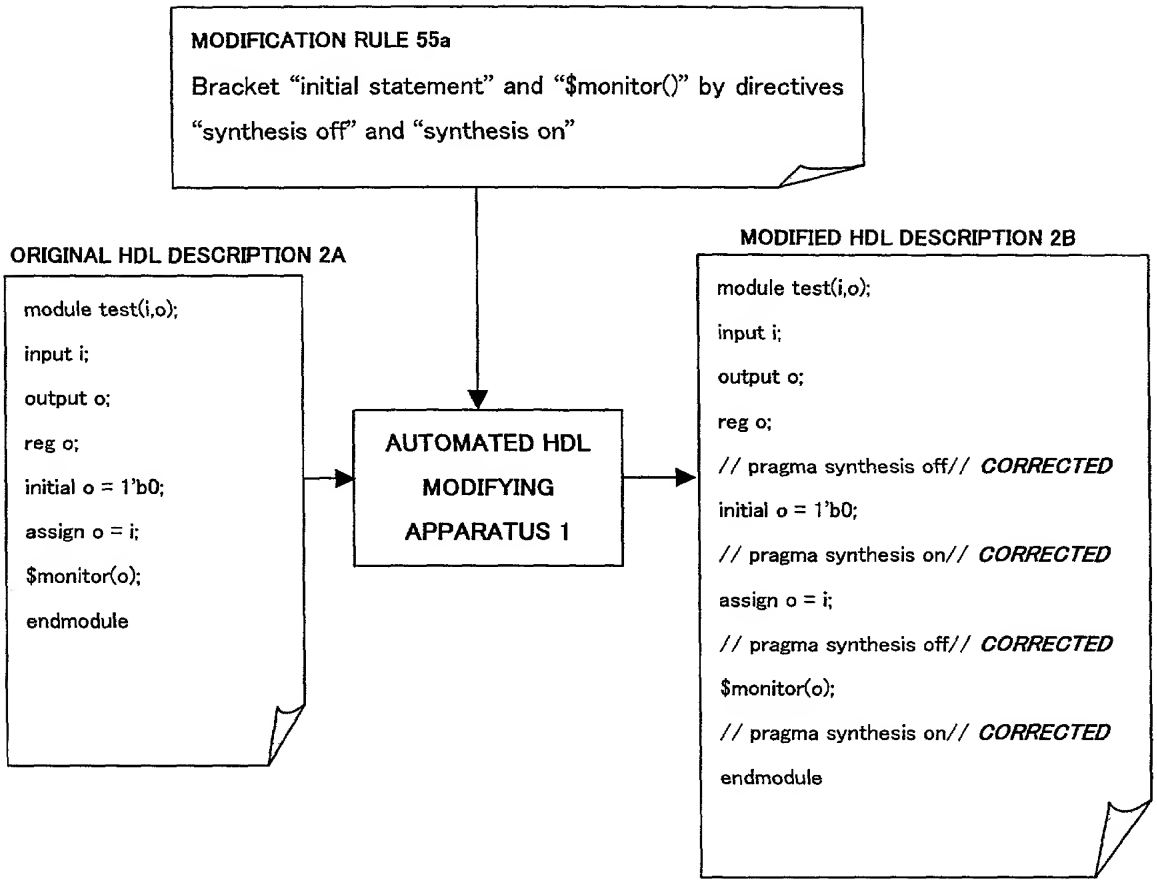
# FIG. 7


MODIFICATION RULE 53c
component port: complement to entity;
ORIGINAL HDL DESCRIPTION 2A
library IEEE;
use IEEE.std_logic_1164.all;
entity TEST1 is
port ( A, B : in std_logc;
C : out std_logc);
end TEST1;
architecture SAMPLE of TEST1 is
component TEST2
port (S,T : in std_logic;
U : out std_logic);
end component;
begin
T2: TEST2 port map( S=>A,
T=>B,U=>C);
end;
AUTOMATED HDL MODIFYING APPARATUS 1
MODIFIED HDL DESCRIPTION 2B
library IEEE;
use IEEE.std_logic_1164.all;
entity TEST1 is
port ( A, B : in std_logc;
C : out std_logc);
end TEST1;
architecture SAMPLE of TEST1 is
component TEST2
port (S,T : in std_logic;
U,V : out std_logic);// CORRECTED
end component;
begin
T2: TEST2 port map( S=>A,
T=>B,U=>C,V=>OPEN);// CORRECTED
end;
library IEEE;
use IEEE.std_logic_1164.all;
entity TEST2 is
port (S,T : in std_logic;
U,V : out std_logic);
end TEST2;
architecture SAMPLE of TEST2 is
begin
U <= S and T;
V <= S or T;
end;
library IEEE;
use IEEE.std_logic_1164.all;
entity TEST2 is
port (S,T : in std_logic;
U,V : out std_logic);
end TEST2;
architecture SAMPLE of TEST2 is
begin
U <= S and T;
V <= S or T;
end;

# FIG. 8


MODIFICATION RULE 53d
port connection: name ;
ORIGINAL HDL DESCRIPTION 2A
module test1(a,b);
input a;
output b;
assign b=a;
endmodule
module test2(p,q);
input p;
output q;
test1 test1_ins( p,q);
endmodule
AUTOMATED HDL MODIFYING APPARATUS 1
MODIFIED HDL DESCRIPTION 2B
module test1(a,b);
input a;
output b;
assign b=a;
endmodule
module test2(p,q);
input p;
output q;
test1 test1_ins(.a( p),.b(q));// CORRECTED
endmodule

# FIG. 9


MODIFICATION RULE 55a
Bracket "initial statement" and "$monitor()" by directives "synthesis off" and "synthesis on"
ORIGINAL HDL DESCRIPTION 2A
module test(i,o);
input i;
output o;
reg o;
initial o = 1'b0;
assign o = i;
$monitor(o);
endmodule
AUTOMATED HDL MODIFYING APPARATUS 1
MODIFIED HDL DESCRIPTION 2B
module test(i,o);
input i;
output o;
reg o;
// pragma synthesis off// CORRECTED
initial o = 1'b0;
// pragma synthesis on// CORRECTED
assign o = i;
// pragma synthesis off// CORRECTED
$monitor(o);
// pragma synthesis on// CORRECTED
endmodule